Paraphrasing controlled English texts

Kaljurand, K

Abstract: We discuss paraphrasing controlled English texts, by defining two fragments of Attempto Controlled English (ACE): Core ACE and NP ACE. We show that these fragments have features that one would usually expect from paraphrases. We also describe a tool that paraphrases ACE sentences into these fragments.

Paraphrasing controlled English texts

Kaarel Kaljurand

Institute of Computational Linguistics, University of Zurich
kaljurand@gmail.com

Abstract. We discuss paraphrasing controlled English texts, by defining two fragments of Attempto Controlled English (ACE): Core ACE and NP ACE. We show that these fragments have features that one would usually expect from paraphrases. We also describe a tool that paraphrases ACE sentences into these fragments.

1 Introduction

An editing tool for controlled natural language (CNL) documents can support its user in various ways, for example pinpoint the location of syntax errors, highlight the resolution of anaphoric references, inform about the semantic contribution of the added sentences (e.g. will the complete text remain logically consistent?, is the addition logically redundant with respect to the complete text?). These various forms of feedback, given by the machine to the human user, help the user to better understand the text that is being created. Thanks to the formal nature of CNLs, such rich syntactic and semantic feedback can be computed automatically.

This paper discusses one form of semantic feedback, namely paraphrasing. A paraphrase of a text is its reformulation (in the same language) such that the meaning of the text is preserved. A paraphrase can be a useful form of feedback for both novice and experienced CNL users — it can clarify the meaning of the text, it can propose a more readable form of the text, it can reveal spelling mistakes, it can help the developer of the CNL parser to detect bugs in it, etc. In order to make paraphrasing possible, the language must contain syntactic sugar, i.e. allow syntactically different forms to have the same meaning.

This paper discusses paraphrasing of Attempto Controlled English (ACE) [2] texts. The paraphrasing is based on verbalizing the discourse representation structure (DRS) [3] — the meaning representation — of the input ACE text in two different fragments of ACE — Core ACE and NP ACE. We extend the work on paraphrasing and DRS verbalization presented in [4] and [1].

Core ACE and NP ACE are two syntactically non-overlapping fragments of ACE. Core ACE covers semantically almost all of ACE, and uses *if-then*-sentences to represent implications. NP ACE covers only certain forms of implications, but uses *every*-sentences to represent them. Paraphrasing into both of these fragments has been implemented in SWI-Prolog and is available under the LGPL license as part of the Attempto Parsing Engine (APE) distribution[1].

[1] http://attempto.ifi.uzh.ch/site/downloads/

2 Requirements

In addition to having an equivalent meaning, we can think of the following requirements that paraphrases could satisfy. Some of them are debatable and not all of them are strict requirements. Also, it would be impossible to satisfy all of them because some of the requirements are conflicting.

The paraphrase must be syntactically different from the original. In case the paraphrase matches the original exactly, it is obviously not useful to the user. In some cases it might be desirable for the paraphrase to differ only in certain aspects and leave the rest of the original text intact. For example, sentence 2 is a paraphrase of sentence 1, such that only the pronominal references are replaced, but e.g. the passive is not changed into active nor sentence coordination replaced by two sentences.

(1) Mary is liked by John and she likes him.
(2) Mary is liked by John and Mary likes John.

Similar sentence structure can help the user to better relate the paraphrase to the original.

The paraphrase must not "leave" the original ACE language, i.e. it should not use any meta constructs such as color and font size, the semantics of which the user would have to learn extra. Again, in some cases it might be desirable to use such constructs but the result would not technically be a paraphrase.

The paraphrase language must be a clearly defined and a syntactically small fragment of the full ACE language, so that the user can first learn the rules of the fragment and can then "pick up" the rules of the full language by looking at the paraphrases of its sentences.

The paraphrase language must not contain any syntactic sugar. Paraphrasing can thus be seen as "normalization" of the original text into a canonical form. Writing texts in such a paraphrase language might be awkward as short-hand constructs are missing and general constructs have to be used instead. This however is not a shortcoming as paraphrases are mainly meant for reading and not for writing/modification. The paraphrase must be easy to understand, not necessarily convenient to write.

The paraphrase must improve readability, e.g. simplify the original sentences, or make them more concise. For example, sentences with object relative clauses are harder to read than sentences expressing the same meaning using passive.

(3) Every book is a document that an author who a publisher likes writes.
(4) Every book is a document that is written by an author who is liked by a publisher.

The paraphrase must teach the interpretation rules of the language. For example, that the determiner 'a' corresponds to an existential quantifier

(5) A dog is an animal.
(6) There is a dog. The dog is an animal.

and that 'every' corresponds to a universal quantifier and can be alternatively expressed by an *if-then*-sentence

(7) Every dog is an animal.
(8) If there is a dog then the dog is an animal.

and that anaphoric references pick the closest antecedent

(9) There is a country. It borders Estonia. Which country is it?
(10) There is a country X. The country X borders Estonia. Which country is Estonia?

3 Core ACE and NP ACE

We define two fragments of ACE that meet several of the requirements listed above. **Core ACE** has the following restrictions as compared to full ACE: there are no relative clauses; no coordination of verb phrases; conjunction of only adjectives, adverbs, and noun phrases; no personal (possessive) pronouns; only existential determiners (i.e. omitted are: 'every', 'for every', 'no', etc.); no negation, 'must' nor 'can' of verb phrases and relative clauses; no support for Saxon genitive; and no passive verbs. It is the case that removing these constructs from ACE does not affect the semantic expressivity of the language because alternative constructs remain, e.g. relative clauses and various forms of coordination can be expressed using sentence coordination. The resulting fragment does not contain any syntactic sugar and in this sense forms the "core" of ACE. Still, Core ACE is not semantically as expressive as full ACE because support for some constructs (e.g. *each of* plurals) is missing. For example, sentence 8 is the Core ACE paraphrase of sentence 7.

NP ACE (where "NP" stands for "noun phrase") focuses only on sentences that express implications. Such expressivity is also the focus of many rule and ontology languages (e.g. RuleML, OWL). Implications in NP ACE are expressed by noun phrases that are quantified by 'every' or 'no', thus achieving a more compact representation compared to Core ACE which uses *if-then*-constructs to express implications. NP ACE does not overlap syntactically with Core ACE as the latter does not allow the 'every' and 'no' quantifiers. For example, sentence 7 is the NP ACE paraphrase of sentence 8. Another example of an NP ACE sentence is sentence 4.

4 Paraphrasing by DRS verbalization

We define paraphrasing as a function $f : t_1 \rightarrow t_2$, where t_1 and t_2 are ACE texts (t_2 is a paraphrase of t_1) that have the same meaning m, i.e. $m(t_1) \approx m(t_2)$. For $m(t)$ we take the DRS representation of the text t, as specified in [3] and assigned by APE. The equivalence ($\approx$) of two DRSs can be defined in various ways, but in the context of paraphrasing we use structural equivalence, i.e. DRSs are equivalent if after suitable reordering of DRS conditions in the same DRS and renaming discourse referents, they can be made lexically equivalent (i.e. equivalent as strings).

We define the paraphrase function f as $f_2 \circ f_1$, where f_1 is the translation of an ACE text into the DRS, and f_2 is translation of a DRS into an ACE text.

f_2 must satisfy the requirement that $f_1 \circ f_2$ is an identity transformation where identity is DRS structural equivalence.

In the following we describe two translations that can play the role of f_2. Both of these functions accept a fragment of the DRS language as input and target a fragment of ACE as output.

The **Core ACE verbalizer** applies a relatively direct transformation of DRS conditions into ACE sentences: *predicate*-conditions (i.e. conditions that correspond to verbs and their complements) map to simple ACE sentences, and embedded DRSs into complex sentences (e.g. negated or *if-then*-sentences). The order of sentences that originate from the same DRS is fixed so that sentences that mention the same nouns are positioned next to each other (in the conjunction). This will result in easier to read sentences.

The **NP ACE verbalizer** is only applied to DRS implications which furthermore must share at least one discourse referent between the *if*-box and the *then*-box. Only such implications can be expressed as *every*-sentences. The *predicate*-conditions in both the *if*-box and the *then*-box are "rolled up" starting with the condition that contains a shared discourse referent. The resulting structures are directly mapped to noun phrases that are possibly modified by (a coordination or negation of) relative clauses.

We tested the coverage of both verbalizers (as neither covers the complete DRS language) on the APE regression test set. The APE regression test set reflects all aspects of the ACE language focusing on sentences and text snippets that are frequently used in knowledge representation tasks, but also including many very complex constructs which are not likely to be used in the real world but are needed to test the parser. The regression test set contains 3283 test cases — mappings from an ACE text to its DRS. Out of these, 862 are negative test cases, i.e. non-valid ACE texts that (must) map to empty DRSs.

We applied both verbalizers to the non-empty DRSs (2421 tests) to check if the $f_1 \circ f_2$ is an identity function. For the Core ACE verbalizer, the roundtrip succeeded in 2134 cases (88%), but in 202 cases the paraphrase was identical to the original. For the NP ACE verbalizer, the roundtrip succeeded in 505 cases (21%) and in 67 cases the paraphrase was identical to the original.

5 Discussion

The described subsets and the paraphrasers still suffer from often mapping an input to itself. For simple sentences (John likes Mary.) this might not matter because the users correctly understand these sentences and would not depend on the paraphrase. There are sentences however that are often misunderstood, notably sentences that feature prepositional phrases (PPs), such as

(11) Every airline charges a passenger with an overweight-luggage.

According to the ACE interpretation rules, the PP in sentence 11 semantically attaches to the verb 'charges' which the user probably did not intend. In our view, the most effective clarification in such situations would not involve a

paraphrase. Rather, the editing tool could detect the PP in this sentence and pop up a tooltip (which the advanced users can disable) informing the user about the ACE interpretation rules that cover PPs. Alternatively, the syntax tree of this sentence could be shown to the users as it makes the attachment structure explicit. For example, AceWiki[2] offers a user-friendly representation of syntax trees, called "syntax boxes".

One type of problem that both paraphrasers currently poorly deal with has to do with the scopes of ACE constructs. For example, the sentence

(12) {Every dog is an animal} or {there is a cat}.

contains a sentence disjunction. (The scopes of the arguments are denoted by curly brackets.) The Core ACE paraphraser replaces the *every*-sentence with an *if-then*-sentence, "forcing" there is a cat to be in the scope of the *then*-part, i.e.

(13) If there is a dog X1 then {{the dog X1 is an animal} or {there is a cat}}.

This however does not reflect the meaning of the input sentence. Failure to deliver a semantically correct paraphrase can be automatically detected by checking if the DRSs of the input and of the paraphrase are equivalent as defined in section 4. The question still remains if a paraphrase can in all cases explain the scoping rules of ACE sentences, or a meta-construct (such as curly brackets) is required and/or more useful as the means of such explanation.

6 Conclusions

This paper demonstrates that ACE has two non-overlapping fragments — Core ACE and NP ACE — that are suitable for paraphrasing ACE sentences. Both fragments have a relatively simple definition and cover constructs that are important for knowledge representation.

This paper also shows how to implement a paraphraser that verbalizes a DRS in either Core ACE or NP ACE. As these fragments do not overlap syntactically, one can choose the paraphrase that is different from the original.

References

1. Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Deliverable I2-D9. Attempto Controlled English 5: Language Extensions and Tools I. Technical report, REWERSE, 2006. 28 pages, `http://rewerse.net/deliverables.html`.
2. Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Attempto Controlled English for Knowledge Representation. In *Reasoning Web, 4th International Summer School 2008, Tutorial Lectures*, 2008.
3. Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Discourse Representation Structures for ACE 6.0. Technical Report ifi-2008.02, Department of Informatics, University of Zurich, Zurich, Switzerland, 2008.
4. Norbert E. Fuchs, Kaarel Kaljurand, and Gerold Schneider. Deliverable I2-D5. Verbalising Formal Languages in Attempto Controlled English I. Technical report, REWERSE, 2005. 21 pages, `http://rewerse.net/deliverables.html`.

[2] `http://attempto.ifi.uzh.ch/acewiki/`